

## AMENDMENTS TO THE CLAIMS

This listing of claims replaces all prior versions, and listings, of claims in the application:

### Listing of Claims:

1. (Currently Amended) A computer-readable storage medium having computer-executable instructions stored thereon that, when executed by a processor, implement a method for associating original source code with binary code for debugging the binary code, the method comprising:

storing a source code file on a server, the source code file including source code and being associated with a version;

compiling the source code file into a binary file;

while compiling the source code file, extracting information that identifies a location of the source code file, the version associated with the source code file, a name of the server, a port of the server at which the server may be accessed to access the source code, a path to the source code, and a numeric value that indicates a version number of the source code;

storing the extracted information in a debug file associated with the binary file, wherein storing the extracted information includes storing the debug file with information from a data stream, wherein the data stream is divided into the following:

a first one or more lines defining a global variables area;

a second one or more lines defining a local variables area; and

a third one or more lines defining a source files area;

after compiling the source code file, receiving an instruction for a debugger to debug the binary file;

after receiving the instruction for the debugger, using the extracted information in the debug file, locating the source code file and associate it with the binary file; and

thereafter debugging the binary file with full source code support by correlating lines of the source code file with binary instructions in the binary file, the source code file including only the source code originally used to compile the binary file.

2. (Previously Presented) The computer-readable storage medium of claim 1, further comprising:

extracting the information from the debug file;

requesting the source code associated with the version from the server via the information;

placing the source code in a directory used by a debugger to debug the executable code;

and

executing the debugger and matching an instruction in the executable code to an instruction in the source code.

3. (Previously Presented) The computer-readable storage medium of claim 1, wherein the source code file includes programming statements which, when compiled, produce executable code in the form of the binary file.

4. (Previously Presented) The computer-readable storage medium of claim 1, wherein the server comprises a version control server that stores a plurality of versions of the source code.

5. (Cancelled).

6. (Previously Presented) The computer-readable storage medium of claim 1, wherein the binary file includes code that was compiled from a plurality of source code files, each source code file associated with a version.

7. (Previously Presented) The computer-readable storage medium of claim 6, further comprising obtaining additional information that identifies the versions associated with the plurality of source code files to the server and storing the additional information in the debug file.

8. (Previously Presented) The computer-readable storage medium of claim 1, wherein the debug file comprises a program database file that is separate from the executable code.

9. (Previously Presented) The computer-readable storage medium of claim 1, wherein the debug file comprises a portion of an executable file that includes the executable code.

10. (Previously Presented) The computer-readable storage medium of claim 1, further comprising

- iterating each source code file that is part of a compilation, each source code file having a version;

- obtaining information that identifies the version of each source code file to the server and a local name of each source code file;

- storing the information in a lookup table; and

- extracting, from the binary file, local names of the source code files that were used in compiling the binary file; and

- for each source code file that was used in compiling the binary file, looking up the version in the lookup table by using the local name of the source code file.

11. (Currently Amended) A system for associating original source code with binary code for debugging the binary code, comprising:

a compiler arranged to compile source code files into a binary file and to generate debug data, source code files including source code and being associated with a version;

a version control server arranged to store versions of the source code files;

an extractor arranged to operate in parallel with the compiler and extract information that identifies a location of the source code, the version of each source code file used to create the binary file, and key values that include at least a name of the version control server, a port of the version control server at which the version control server may be accessed to access the source code files, a path or paths to the source code files, and a plurality of numeric values, each numeric value indicating a version number of a corresponding source code file, wherein the extractor is further arranged to store the extracted information in a debug file for use in retrieving the source code files at a debug time, wherein storing the extracted information includes storing the debug file with information from a data stream, wherein the data stream is divided into the following:

a first one or more lines defining a global variables area;

a second one or more lines defining a local variables area; and

a third one or more lines defining a source files area; and

a debugger arranged to, after compiling of the source code, receive an instruction to debug the binary file and use the extracted information from the debug file to locate the source code file and associate it with the binary file, and thereafter debug the binary file with full source code support by correlating lines of the source code file with binary instructions in the binary file, the source code file including only the source code originally used to compile the binary file.

12. (Previously Presented) The system of claim 11, further comprising a source server arranged to extract the information at debug time, retrieve the source code files from the version control server, and place the source code files in a directory accessible by the debugger.

13. (Previously Presented) The system of claim 12, wherein the source server comprises a component of the debugger.

14. (Previously Presented) The system of claim 12, wherein the source server is separate from the debugger.

15. (Original) The system of claim 14, wherein the debugger is arranged to find the source code files in the directory and is unaware of the version control server.

16. (Previously Presented) The system of claim 11, wherein the extracted information comprises key values including a name of the version control server, a port of the version control server at which the version control server may be accessed to access the source code files, a path or paths to the source code files, and a plurality of numeric values, each numeric value indicating a version number of a corresponding source code file.

17. (Cancelled).

18. (Currently Amended) The computer-readable medium of claim 2, wherein placing the source code in a directory used by the debugger to debug the executable code is performed after launching the debugger, ~~and wherein wherein the method further includes: storing in the debug file information from a data stream, and wherein the data stream is divided into a global variables area, a local variables area, and a source files area, the data stream including includes variables in~~ as shorthand expressions requiring an extractor substitute the variables with an expression or value identifying information necessary for retrieval of the source code file, and wherein the method further includes; and merging the debug file with the binary file.

19. (New) The computer-readable medium of claim 1, wherein each of the global variables area, local variables area, and source files area are separated by a respective SRCSRV: line.

20. (New) The computer-readable medium of claim 1, the data stream defines substitute variables for expressions, and wherein each of the global variables area, local variables area, and source files area has a variable assigned to an expression by separating each variable and each expression with an equal sign.